Victorian 6502 User Group Newsletter

KAOS

For People Who Have Got Smart

Registered by Australia Post
Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol.5 No.1 October 1984

MORE FACES FROM KAOS

From left:

standing
George Nikolaidis
Warren Schaeche
Ian Eyles
Ron Kerry (rear view)
Christine Eyles

seated
Michael Lemaire
Paul Dodd

The next meeting will be on Sunday the 28th October at 2pm at the Essendon Primary School which is on the corner of Raleigh and Nicholson Streets, Essendon. The school will be open at 1pm.

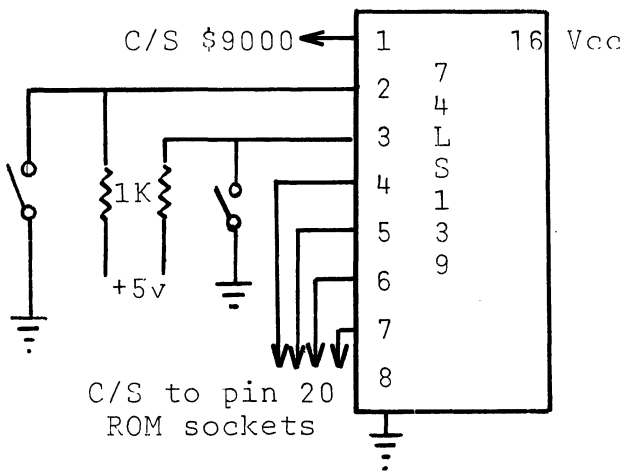
The closing date for articles for the November newsletter is the 9th of November.

INDEX

Apple Disk II Interface 14
Braille and Computers 3
C1P in Candlelight 12
CP/M File 13
Extra ROM 2
For Sale 15
Forth - Understanding pt 4 7
Help Wanted 15
MDMS - More on 12
Meeting - KAOS 11
Meeting - NSW 3
Meeting - QLD 13
Meeting - WA 14
Random Access Files 4
ROM Integrity Test 6
Super Doodler - Review 6
SUPERBOARD 4

EXTRA ROM FOR THE RABBLE EXPANSION BOARD
by Clive Harman

If you are using the Superboard witn the Rabble Expansion Board and have an Eprom Burner available, you may have thought of using all the spare ROM sockets on the board that have addresses normally decoded for the Superboard, such as the BASIC ROM and the Keyboard. Here is an easy way to get the use of those spare sockets.



Install a 74LS139 on the proto area of the Expansion Board. Cut the tracks to pin 2Ø of the sockets for $9ØØØ, $AØØØ, $BØØØ, $DØØØ. This is assuming you wish to have the $CØØØ socket available for Scratch RAM at $C8ØØ. Pick up the CS line for $9ØØØ from a suitable point near IC13 on the decoding area, connect to pin 1 of the 74LS139. Connect pins 4,5,6,7 of the 74LS139 to pins 2Ø of $9ØØØ, $AØØØ, $BØØØ, $DØØØ. Connect pins 2 & 3 of the 74LS139 to a miniature DIP switch on the proto area. Have them held high through a 1K resistor and switched low by the switch. Depending on the switching combinations used all those spare ROM sockets are now available as ROM at $9ØØØ.

If you do not wish to use a switch but prefer to be a little more sophisticated, write a small routine to use the PIA to select pins 2 & 3 high or low to achieve the same purpose. The ROM select can then be Menu driven to select the chip required. If you do this you could put the routine in the Disk Bootstrap area of the Monitor ROM if you intend to use the system for cassette only or you could put a small routine in $8ØØØ.

Which ever system you decide to use you can have a lot of fun devising your own Eproms with all sorts of routines. If you have the $EØØØ Eprom from John Whitehead you can easily relocate BASIC programs to $9ØØØ and run them there from Eprom. John also has available utility routines for Eprom including Renumber and Trace routines (which really work); Hex/Dec converters etc. There is available a revised Assembler that runs at $8ØØØ/$9ØØØ. John Whitehead has the WP65Ø2 available in Eprom for $9ØØØ. You could write some small sound routines for the Sound Generators and have them in Eprom to use as subroutines with existing games. With an extra 12K of Eprom readily available, all sorts of extra routines can be stored for immediate use.

While you are at it, burn BASIC ROMs in 2716s. With the addition of the BASIC 3 fix and the replacement BASIC 4 from Ed Richardson in Qld., I think a very powerful non Disk unit could be developed to give the old Superboard new life as a cheap unit.

All we need is for some "Bright Spark" to develope a PC board for the 6545 screen video mods Bernie Wills keeps developing. How about it fellas, I'll place an order for two now!!

BRAILLE, COMPUTERS AND THE BLIND

by Jeff Kerry

Blind people are rather up against it when it comes to dealing with computers, as conventionally the outputs are mostly visual, e.g. VDU's, printers etc.

The Victorian Education Department policy is that, where possible, handicapped students should attend normal schools instead of special schools that cater for their disability. Blind students in a normal school have an extra disadvantage, since most schools these days (such as the one where I teach) run Computer Studies classes for almost all students, and are increasingly using computers as a classroom learning aid.

In an attempt to cope with this situation, I've been developing some circuits and software to make the computers more accessible to blind students. A speech synthesiser was the first step, along with an interface between the computer and the student's mechanical Brailling machine.

The software enables translation of the Braille into printed ASCII text and speech simultaneously. The hardware is driven from the 16 pin I/O bus (described in KAOS V4/4), modified slightly to suit an Apple, which most schools are already equipped with. The prototype hardware includes a card which plugs into any slot, and an external board which provides about fifty programmable input or output control lines (which could have many other uses.)

Other software under development includes a sort of blind person's talking word processor, and a reverse translation system whereby ordinary text typed in normally is translated into Braille, and printed out on a standard Epson-type printer as a black-and-white dot pattern. A process is now available by which we can, in minutes, convert this into a sheet of raised tactile Braille, using ordinary school office equipment.

This is proving a great time saver for teachers and for visiting specialists, who usually get the tedious manual task of translating students' Brailled work for teachers to read, and converting teachers' work into Braille for the student. The computer is eminently suitable for this sort of translation process, and the results are easily readable by our resident blind Braille experts!

I'd welcome any enquiries, information or ideas on this project - please call me on weekends. Many thanks to KAOS members for support and information, particularly regarding the 16 pin I/O bus. The prototype was operating and on display at the last KAOS meeting, and probably will be at the next.

KAOS NSW

At our last meeting there were 7 members and 3 computers. As this scribe was away no activity report is available.

The next bi-monthly meeting will be held in the Lugarno Girl Guides Hall on Sunday October 28th at 10am. If you haven't been for some time now is the time.

It is planned that the December meeting will be held on Sunday 16th December. All members are requested to make an early note of this date so that they can attend for some if not all of the day.

For information contact N. Bate 02 53 8169 or N. Bissett 02 411 7142

Superboard

October 1984.

Newsletter of the Ohio Superboard User Group, 146 York Street, Nundah, 4012.

RANDOM ACCESS FILES UNDER OS-65D

Random access files are used whenever data needs to be accessed by number or be modified at some later date. Such applications as membership numbers, account numbers, part numbers and etc., are prime examples of their use. 65D has very limited use of disk files, and the random access file in particular leaves a great deal to be desired.

(1) Records are of fixed length. KAOS printed two pokes which extend the range but you are still limited to powers of two.
(2) The DOS will read a track into the buffer every time you request a record, even if the particular record is already there. This causes unnecessary delay and wear on drives and disks. 65D 3.3 has fixed this problem.
(3) Twelve pages are stored on an 8" disk track when thirteen pages are available.
(4) A carriage return and line feed are stored after each record, even though the line feed is not used. If your records are short, then a significant portion of disk space contains useless information.
(5) There is no facility to use a data disk. This means the tracks which normally hold the system are not accessable for use as data files. This is particularly important in 5¼" systems when a lot of data is to be stored.

The following program solves all these problems, and works with either 5¼" or 8" disk systems with one small change. It was designed for use on a 65D 3.2, and would need modification to work with 3.3. By changing the variables in a single line, and re-writing some input and print statements almost any application can be catered for. The program has been written so that it will run on any OSI disk system regardless of disk or memory size. However this necessitates inefficiencies which can easily be remedied by making a few changes to set it up specifically to suit your system.

As written, the program allows for a thirteen page buffer below the BASIC program, which is stored along with the program on the disk, wasting a track. Changes for a 40k C1P with 5¼" drive would be:-

Ignore line 10 REM. Change to 10 POKE 133,151:POKE132,255:CLEAR
line 25 change variable M to 152
line 310 and 320 change 33ØØ to 98ØØ

This moves the buffer to the top of memory. In addition, all remarks and the test and set-up routines from line 499 on, can be deleted from the working program once used. Do not delete line 999.

```
10 REM Before typing in program, POKE12Ø,1:POKE121,64:POKE16384,Ø:NEW

20 POKE2Ø73,96:POKE2893,28:POKE2894,11:REM No CR or CTRL C exits allowed.
25 M=51:P=256:OL=91Ø5:OH=91Ø6:IL=9Ø98:IH=9Ø99
30 RL=16:NR=1ØØØ:FT=1:NP=13:DD=1:DD$="A":REM See variable list,important!
35 RT=INT(NP*P/RL):CR$=CHR$(13):NP$=STR$(NP):IF NP=13 THEN NP$="D"
40 PRINT"Insert data disk #"DD" in drive "DD$:INPUT"READY";A$
45 DISK!"SE "+DD$
50 PRINT:PRINT"To end the program,use the word  END"
```

```
60 PRINT:INPUT"Record Number";RN$:IFRN$="END"THEN999
70 FORR=1TOLEN(RN$):A$=MID$(RN$,R,1):IFA$<"0"ORA$>"9"THEN50
80 NEXT:RN=VAL(RN$):IFRN>NRTHENPRINTNR"Records only":GOTO50
90 GOSUB 300:GOSUB 400
100 PRINT:INPUT"Want to change it";A$:IF ASC(A$)<>89 THEN 50
120 PRINT:INPUT"New Record";R$
130IFLEN(R$)>RL-1 THENPRINT"Maximum is"RL-1"characters":GOTO120
140 GOSUB 200:PRINT"Recorded":GOTO 50
199 REM * Write new record to buffer *
200 POKE OL,L:POKE OH,M+H
210 PRINT#5,R$;CR$;:PRINT#9:WF=1:RETURN
299 REM * Record READ/WRITE/FIND routine *
300 NT=FT+INT(RN/RT):IF NT=PT GOTO 330
310 IFWF=1THENDISK!"SA "+RIGHT$(STR$(100+PT),2)+",1=3300/"+NP$:WF=0
320 PT=NT:DISK!"CA 3300="+RIGHT$(STR$(100+PT),2)+",1"
330 B=(RN-(PT-FT)*RT)*RL:H=INT(B/P):L=B-H*P:RETURN
399 REM * Display record *
400 POKE IL,L:POKE IH,M+H
410 INPUT#5,R$:PRINT"Record #"RN;R$:RETURN
499 REM * Test write all files * Insert line 47 GOTO 500
500 FORNT=FTTOFT+INT(NR/RT):DISK!"PU "+RIGHT$(STR$(100+NT),2):NEXT
510 FORRN=0TONR:GOSUB300:R$=STR$(RN):GOSUB200:NEXT
520 PRINT"Test data complete : Delete line 47":GOTO999
598 REM Test read all files. Insert line 47 GOTO600
600 FORRN=0TONR:GOSUB300:GOSUB400:NEXT:END
999 GOSUB310:DISK!"SE A":POKE2073,173:POKE2893,55:POKE2894,8:END
```

Variable list in order of appearance:-

- M Memory location for buffer start
- P Page size
- OL,OH Output to memory LO/HI byte pointers
- IL,IH Input from memory LO/HI byte pointers
- RL Record length. Can be anything from 2 bytes to the limit of the track.
- NR Number of records total. Can always be increased later if tracks are free
- FT First track used. Don't use your system disk with FT=1
- NP,NP$ Pages per track. Use 8 for 5¼" disks, 13 for 8" disks
- DD Data disk number. Used only as a prompt
- DD$ The drive you planned to use DD in
- RT Records per track
- CR$ A carriage return must terminate each field of the record
- RN$,RN Record number, or way out of the program
- NT Needed track to be called into buffer
- PT Present track in buffer. If NT=PT, you don't need to read the disk
- WF Write flag. Only if records have been updated does the disk write
- B Bytes into buffer where the required record is stored
- L,H Bytes converted to LO byte/HI byte for the memory pointers

Well, that's about it. If your requirement is more than one field, then remember that a CR$ must be stored after each field of the record, and that the record length will be the total lengths of all the fields plus one CR$ for each field used. Otherwise one record can overwrite the next one. OSI's system had that problem also.

- -

NEXT MONTH: Alan Cashin's M/C string sort routine - for Disassemblers Anonymous. Also a BASIC 3 fix considered to be superior to the one published by Earl Morris in KAOS.

— SUPERBOARD —

SIMPLE ROM INTEGRITY TEST

The OSI BASIC and SYNMON roms are mask programmed devices, and these chips sometimes develop faults after a number of years due to an effect known as silver migration. The chip is not destroyed by this process, and the defect can usually be rectified by re-programming. However I doubt if anyone in Australia owns a programmer that would do it. The simplest approach is to replace the chip with a 2716 Eprom, and modify the chip select lines on the computer slightly to suit.

Now integrity is defined in my dictionary as "Original perfect state" and I doubt that Microsoft's original effort could be called that, however the following routine will allow you to test your roms. Although the test is very rough and ready, the chance of getting a correct result if the rom is in fact faulty, is less that Ø.4%.

The routine could also be modified for Eprom, and stored somewhere in your system and called on every reset to check the rom integrity. A message could be printed if there was disagreement between actual and expected results.
The routine as below lives in the input buffer area for convenience.

```
0034 A0 00     LDY #$00
0036 98        TYA
0037 18        CLC
0038 71 30     ADC ($30),Y
003A 20 4E 00  JSR $004E
003D A6 30     LDX $30
003F E4 32     CPX $32
0041 D0 F5     BNE $0038
0043 A6 31     LDX $31
0045 E4 33     CPX $33
0047 D0 EF     BNE $0038
0049 85 55     STA $0055
004B 4C 00 FE  JMP $FE00
004E E6 30     INC $30
0050 D0 02     BNE $0054
0052 E6 31     INC $31
0054 60        RTS
```

	TEST DATA				RESULT
	0030	0031	0032	0033	0055
BASIC 1	00	A0	00	A8	6B
BASIC 2	00	A8	00	B0	3D
BASIC 3	00	B0	00	B8	65
BASIC 4	00	B8	00	C0	79
SYNMON	00	F8	00	00	C5
1,2,3 & 4	00	A0	00	C0	86

NOTES

Put the test data in the locations as above. Run the program at $0034. After the screen clears, look in location $0055 for the result.

SOFTWARE REVIEW - *Super Doodler*

As the name suggests, Super doodler is a graphics generation program, written by one WT Musser, and formerly distributed by Aardvark. Once you get accustomed to the keys, it is not hard to use. The basic idea is to generate graphics for possible later use in games. Having drawn the picture you can save it on tape as data statements. The documentation that comes with Super Doodler gives a line of BASIC you can use to load in the saved data.
The cursor can be any of those in the OSI character set, and can be moved around the screen in Transparent, Erasing, or Drawing mode. As the cursor moves, the decimal memory location being occupied is displayed at the bottom of the screen.

The program needs 8K to run, and is written in BASIC. A C1P version is in the OSUG library, for library members, at the usual postage costs.

Ed Richardson.

PS. I know the Rom Integrity Test could be written more efficiently.

UNDERSTANDING FORTH part 4
by Ray Gardiner

: MEMORY CONSIDERATIONS

This month we shall examine the words @ ! C@ C! and some special cases like C, and "," , as well as introducing constants and variables. Firstly, the formal definitions,

Forth word	Stack effect	Comments
@ "fetch"	addr ---- data	fetch 16b data at addr
! "store"	data addr --- ...	store 16b data at addr
C! "c store"	Data addr --- ...	store 8b data at addr
C@ "c fetch"	addr --- ...	fetch 8b data from addr
+! "plus store"	inc. addr --- ...	increment data at addr by the amount inc.
, "comma"	data --- ...	store 16b data at HERE
C, "c comma"	data --- ...	store 8b data at HERE
HERE "here"	... --- HERE	return current dictionary pointer
?	addr ---	display data at addr
DP "dictionary pointer"	... --- DPaddr	address of DP

Just for interest's sake here are the definitions of C, , HERE and ?.

```
: HERE DP @  ;      DP is the system dictionary pointer
: , HERE !   ;
: C, HERE C! ;
: ? @ .      ;
```

As you can see the definitions are very short , but none-the-less, very powerful words. To illustrate the power of these simply words consider the following code which might appear in an assembler for the 6502 processor. Written in Forth the single byte opcodes can be written directly.

```
HEX
: TAX AA C, ;
: TAY A8 C, ;
: PHA 48 C, ;
: JSR 20 C, , ;    assemble JSR to address on stack
: JMP 4C C, , ;    assemble JMP to address on stack
and so on.
```

: VARIABLE

The word 'Variable' is a defining word which assigns a name to a memory address. The memory address is left on the stack at run time so that data can be stored ! or fetched @ from the location, the actual location used is irrelevent since all references to the location are made by name.

```
0000 VARIABLE DAY

23 DAY !        set Day to 23
DAY @           leave current value of DAY on the stack
DAY ?           read DAY and print value
```

Variables are used for any data which changes after compilation, all variables defined in this way represent 16 bit signed or unsigned numbers.

: CONSTANTS

A constant differs from a variable only in that when executed, the data is returned, rather than the address where the data is stored, for example the boiling point of water is constant, (for a given pressure at least).

```
100 CONSTANT BOILING.POINT
000 VARIABLE TEMPERATURE

: ?BOILING TEMPERATURE @
           BOILING.POINT = IF    ." coffee time folks " LEAVE
                           ELSE  ." a watched pot etc "
                           THEN ;
: DELAY 8000 0 DO LOOP ;
: HEAT.POT 1 TEMPERATURE +! ;
: COFFEE 150 0 DO 13 EMIT HEAT.POT  DELAY  ?BOILING LOOP ;
```

: LOGICAL OPERATORS

The logical operators are now introduced and their actions described.

Forth word	Stack effect	Comments
AND	n1 n2 --- n3	the bitwise and of n1 n2
OR	n1 n2 --- n3	the bitwise OR of n1 n2
XOR	n1 n2 --- n3	the bitwise exclusive OR

Any other possible logical bitwise operations may be defined using these three. For example

```
: NAND AND 255 XOR ;
: NOR  OR  255 XOR ;
```

: BASE

BASE is a system variable like DP and others. It is used to control all the numeric string conversion for both input and output, consequently it is often referred to as the I/O base. Several Forth words are definned in the standard system for the more popular I/O bases.

```
: DECIMAL 10 BASE ! ;
: HEX 16 BASE ! ;
: BINARY 2 BASE ! ;
```

You may add your own if your application requires a special base for whatever reason, however remember to change back!

Try the following examples.

```
(problem, what is F467 (hex) in binary)
HEX F467 BINARY . DECIMAL

(problem, what is 6 times 9 in base 13)
(or the question to Deep Thought's answer)
13 BASE ! 6 9 * . DECIMAL  (wonder if Douglass Adams knows?)
(a simple hex to decimal utility)
: HEX->DEC  BEGIN
                  CR ." INPUT HEX " HEX QUERY INTERPRET
                     ."=" DECIMAL o D.  (unsigned)
                     ." DECIMAL "
              AGAIN ;
```

Type QUIT instead of a number to exit, otherwise loop forever.

As you no doubt will play around and experiment with BASE we can move on to the more involved aspects of VARIABLE.

: ARRAYS

Arrays are created in the same way as variables, the only difference is that instead of referencing one memory location, the array consists of a block of consecutive locations in memory. Space for an ARRAY is reservd by the word ALLOT which takes a byte count from the stack and advances the dictionary pointer.

```
: ALLOT DP +! ;

ØØØØ VARIABLE TEMPERATURE 18 ALLOT
```

The array created is called TEMPERATURE and sufficient memory has been reserved for 2Ø bytes of data. Why 2Ø? I hear someone ask, when we only ALLOTed 18 bytes extra. The extra two bytes were already reserved by VARIABLE. Needless to say, this array is sufficient for 1Ø x 16 bit variables. To store and retrieve data from this array you simply add an offst to the address returned by the variable TEMPERATURE.

```
: !TEMP  ( temp n --- ) 2 * TEMPERATURE + ! ;
: @ TEMP  (n --- temp ) 2 * TEMPERATURE + @ ;
```

Suppose coffee pot number five was 86 degrees celsius 86 5 !TEMP would write the number 85 into the fifth spot in the array, to read this back at a later stage, 5 @TEMP would do the task of returning the fifth element of the array.

Data structures in Forth are a complete subject on their own, and we shall return to arrays and such when we cover the words which DEFINE defining words. (Think about it.)

: CASE

As promised last month I will provide the source for Dr Eaker's CASE statement. I strongly reccomend that you read the Forth Dimensions article, vol II/3 pp 37-4Ø. The 65Ø2 code for (OF) is by yours truly and follows the guide lines given in the Forth Dimensions article.

```
: CASE ?COMP                          compiling only
       CSP @ !CSP                     save current csp
       4 ; IMMEDIATE                  leave the 4 for syntax checking
```

The following code is nucleus dependant, check your own system! It was written for fig-Forth.

```
HEX ASSEMBLER
CODE (OF) INX, INX, BOT 2 - LDA, BOT CMP,    pop case and cmp lo
      Ø= IF,                                 compile bne
           BOT 1 - LDA, BOT 1+ CMP,          compare high bytes
         Ø= IF,                              compile bne
           INX, INX,                         found a match so
           ' ØBRANCH CFA @ 8 + JMP,          step over and execute
         THEN,                               execute the code
       THEN,                                 test failed so look
      ' BRANCH CFA @ JMP,                    at next case
```

```
: OF 4 ?PAIRS                          check syntax
       COMPILE (OF)                    run-time code
       HERE Ø ,                        space for offset
       5 ; IMMEDIATE                   use 5 for syntax

: ENDOF 5 ?PAIRS                       check syntax
         COMPILE BRANCH                run-time branch to endcase
         HERE Ø ,                      reserve room for offset
         SWAP 2 [COMPILE] THEN         compile THEN to resolve branch
                                       from previous OF
         4 ; IMMEDIATE                 syntax check uses 4's

: ENDCASE 4 ?PAIRS                     syntax checking
         COMPILE DROP                  if no match at run-time
         BEGIN                         resolve all forward branch
               SP@ CSP @ = Ø =         from ENDOF's to point after
         WHILE                         the drop
               2 [COMPILE] THEN        use THEN to resolve branch
         REPEAT
CSP ! ; IMMEDIATE                      restore stack as you were
```

This version of a case statement is simple to use and powerful enough for a large number of applications. As such it illustrates how additional control structures can be easily added to the language. If your mind goes blank when you study the code above, then relax, we ain't covered most of those things yet.

A few hints for the more advanced reader, the word IMMEDIATE toggles on the precedence bit in the name field header so the word just defined will EXECUTE at compile time rather than compile. With this simple method you can define words which control the behaviour of the compiler while it is running. Sometimes you may want to force such a word to compile rather than execute, this is accomplished by the word [COMPILE] XXXX where XXXX is an immediate word.

Answer to last month. RG 9/84

This word dumps memory in hex and ASCII with operator control

```
2Ø CONSTANT P.LEN    (number of lines per page    )
: PRINT.ASCII        (output only printable ASCII characters    )
   127 AND DUP 32 < IF ." ." DROP ELSE EMIT THEN ;

: D.LINE             (dump one line in hex and ASCII     )
   CR DUP U. SPACE
   16 Ø DO DUP I + C @ 3 .R     (loop 2 spaces  )
   16 Ø DO DUP I + C @ PRINT.ASCII LOOP   DROP ;

: CHK.OP  ?TERMINAL IF CR ." dump aborted " QUIT   THEN ;

: D.PAGE DUP P.LEN 16 * + SWAP DO I D.LINE CHK.OP 16 +LOOP ;

: MEM.DUMP BASE C@ >R HEX
           BEGIN DUP
                 D.PAGE CR ." press space to continue "
                 P.LEN 16 * + KEY 32 = Ø=
           UNTIL DROP R> BASE C! CR ;
```

THE MEETING WAS KAOS

by King Corky

You heard in the last newsletter that Compsoft is closing the doors on their shop-front operation, George and Paul, (Michael has gone on to pursue other forms of brain exercise with another company), will however still be operating, (another back-yarder?), and we hope to still be able to get bargain prices for hardware and software, I'd like to remind you that the most flexible CP/M system ever devised, (along with the most flexible disk controller ever devised), was developed by the staff at Compsoft and that even though I know of only one member that has it installed, (I will have mine in soon and Jeff Rae is also planning to install one), it is still great value for money, even for the floppy controller alone.

Nigel Bisset and company, (Sydney), has developed a stand-alone controller running a 65Ø2 with 32 I/O lines, 8K RAM, 8K ROM, that will sell for around $28. He has one in operation running 8 slave devices. This type of unit would be great for household monitoring functions, i.e. doors, windows, swimming pools, alarms, lights, toasters, radios, microcomputers, etc. Who knows, somebody might invent an electrically operated toilet suite that could be hooked-up to the controller, for whatever purposes a devious mind might concoct. It turns your system-bound, peripheral-restricted micro into a totally intelligent system, (I must get one, it will be the the only intelligent thing in my house). The next step is for someone to devise a cheap, (very), FM transmitter/reciever controller setup to allow the remote control of all those functions via the mains. There was such a unit sold in the USA and UK some time bk, (BSR), but finding the controller and slave units is difficult, if they still exist.

David Dodds requires help in sorting the 5.25" disk library. Apparently there is a lot of duplicated software in various formats splashed all over various disks. If you can help, get in touch with David, his number is on the front page.

For quite some time now, Jeff Kerry has been showing us all what a little ingenuity can do, he has produced his own version of the grossly over-priced commercially-produced Turtle, created wonderous variations of 'human' speech and made a monster of a devil, (an Apple clone). He carries his varieties of OSI type micros around in brief cases and quietly goes on achieving more than most could hope to. His latest project is a very worthwhile one, using a micro to allow a blind student to comunicate with a computer by Braille. He is also writing software to translate text into Braille and vice-versa. He also plans to include speech synthesis into the package.

A neat little device was being handed around at the last meeting. A single hole punch for making double sided out of single sided disks. No measuring or pencil lines needed. The disk fits into the punch and is automatically aligned at the correct location for the notch. It usually sells for about $15 from a firm, (I have forgotten who), at 495 High St Rd, Mt Waverly.

Next month I hope to have some information on NOS-BASICODE. This is a translation code, (in BASIC), that ewill allow transfer of programs from almost every micro on the market, (including OSI), to any other, providing that certain rules are followed. The criteria is not an involved one. The main limitation is that all program material is cassette based. The 'foreign' program is loaded into the micro that has first been loaded with the translater which is then RUN. It converts the 'other' program into the syntax required for your particular brand of micro. I have written to the perpetraters in Sweden and should have a reply back by the end of October, (I hope). More info as it arrives. The OSI translater should come with the info as well.

THE C1P IN CANDLELIGHT

by Ken Maclean

Do you have trouble when the power fails? Have you ever typed in your literary masterpiece in WP65Ø2 and, just before you tape it, the dishwasher starts and you lose the b----y lot? Read on - maybe I can help.

In my limited experience of home computers, most standard type power supplies produce a DC voltage at the filter capacitors of the order of 1Ø volts. This is then supplied to the regulator circuit to produce the required 5 volts.

If you use a power transformer which has a slightly higher AC output then the voltage at the filter capacitor can be increased to 12.5 volts. You can now connect a 12 volt car battery to this point and only a direct lightning strike will corrupt your precious program. If you are really keen, if you are using a converted 12" TV as a monitor, this also can be connected to the 12 volt supply. A suitable power transformer for the above purpose is a transformer intended for a battery charger.

So now you have your battery connected and no longer have to worry about the dishwasher starting. After that late night session you switch off the power to the computer before retiring only to find next morning that you had left the battery still connected and that the computer is still running. Ok, no sweat, connect a relay to the low voltage AC from the transformer through a 1 amp diode and use that relay to apply 12 volts to a sonalert to tell you when the power is off but the battery is still on. Now you have the best of both worlds.

MORE on OS-MDMS

by Norman Bate

Since my last article on MDMS (Feb 84) I have written three more utility programs to add to the existing set of MDMS programs.

The programs are:-

(1) Add an extra field to a Master File.
(2) Print out field specifications (screen and/or printer).
(3) Print address mailing labels.

1. This program overcomes a major drawback with already filled master files, in that, at present, if more fields of data are required, then a new master file has to be created and all of the old data re-typed into it. This program eliminates the re-typing. To use this program, you first create a new master file using the same field specifications as before plus the new additional field. Run 'AD1FLD' and the old data is transferred and the new field loaded with 'Ø's' in blocks of 2ØØ records (this can be changed to suit).

2. Field specifications of a master file (if you haven't written them down) can only be retrieved by using the program 'CREATE' (review only) and 'MFDUMP' (screen only). This program offers the same choices as a DIR program (ie screen and/or printer).

3. 'ADRESS' inputs records one at a time from a master file into an array and prints out address labels using the fields 'NAME, STREET, TOWN and POSTCODE' laid out for a run of single labels. If needed, the program could easily be modified to input 1, 2 or 3 records at a time for wider stationery.

A copy of all my MDMS programs has been forwarded to the library on a 5.25" disk. Page copies and/or transfers to your disk can be obtained from me with a SAE to

THE CP/M FILE

This is a call to all those KAOS members who have lashed out and upgraded their machines, OSI or Rabble, to the CP/M operating system.

"Hey, you guys, stand up and be counted".

Seriously, quite a few of you have put out good money for an enhancement which has some pretty powerful specs, especially when you compare it with some of the other offerings on the market at such inflated prices.

There is a bunch of dedicated hackers, who shall remain un-named (better so), who are working to modify software, as well as writing new stuff, so that you can extract even better performance from your 'toy'.

The greater part of this 'midnite oil' is being burnt at no cost to KAOS CP/Mers. However, your active support is required, because as in all human (?) endeavour, you are going to get out of it according to what you put in.

The first thing I would like from all K/C's, both active and potential, is their 'curricula vitae' (for those who don't understand Ancient Italian, that means who, where why etc), so that I can set up dossiers on you lot (this IS 1984).

Secondly, for those of you who are active, how about a catalogue of your software, in particular any which you think might be of interest to other K/C's. If you do this in the form of a CP/M file then it will be easy for us to set up a master catalogue of benefit to all. How about it?

All info and/or queries should go to:-
R.M. HESS

Let's make this the best Special Interest Group in KAOS!

For those of you who are worried about Compsoft's closure in Richmond, C/S still lives. In any case the KAOS CP/MUG will continue to support the system out there for quite some time yet, both software and hardware.

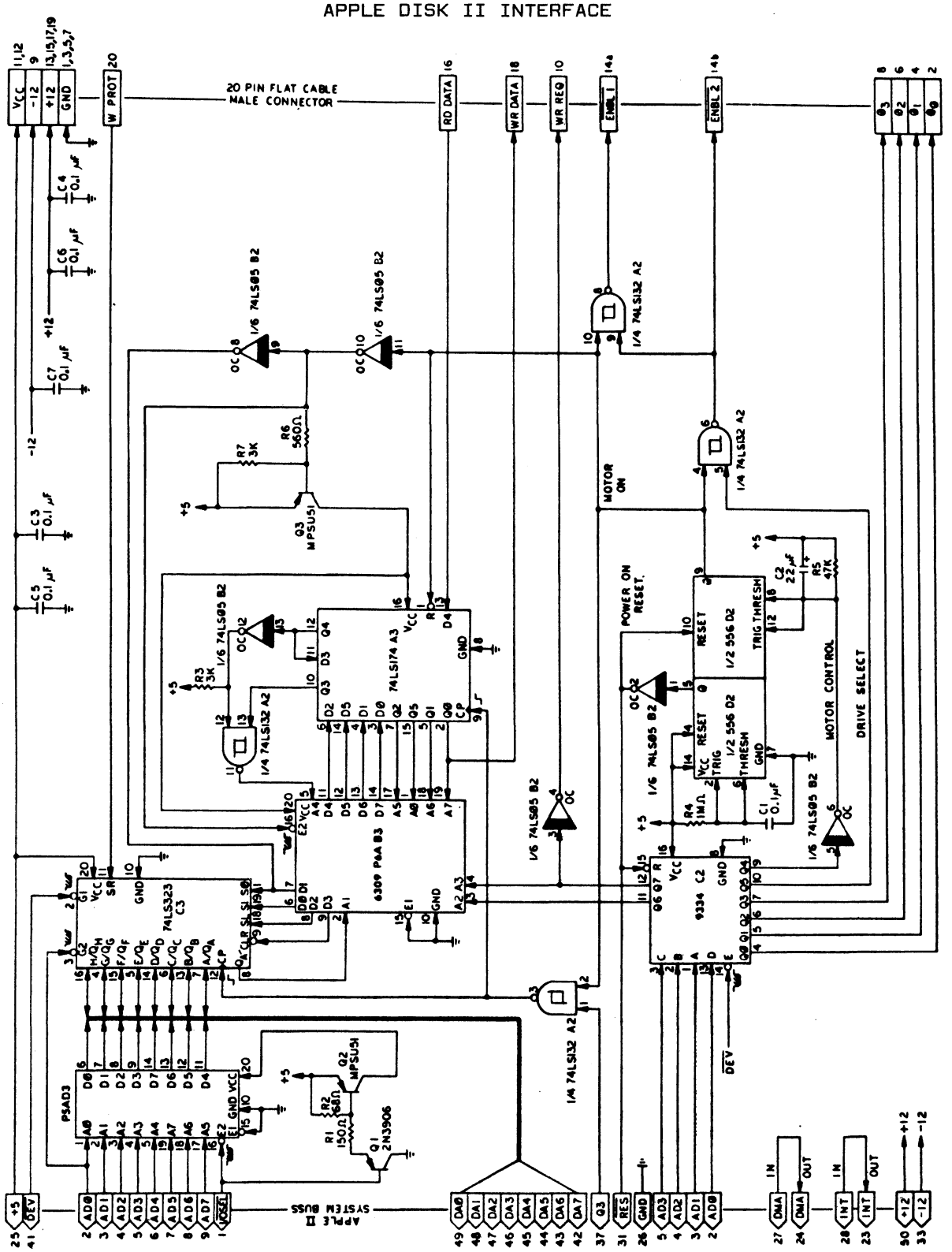
If you have any queries or concerns, please bring them and yourselves to the meeting on 28th October.

KAOS Queensland User Group Meeting 7/10/84

Attendance: 12 Computers: 6
The OSIs and BBCs were evenly matched at three each. There was so much software for the Beebs that Paul Brodie said his main problem was finding enough time to run all the programs. John Froggatt had a voice synthetiser in his machine which was of exceptional quality. Alan Calvert had solved all the remaining problems in his 8" OSI and had ordered a couple of the BASIC-in-ROM DOS's from the U.K. Bernie Wills had his machine in and demonstrated the noise free display mod.

OSI owner Max Bain had sent me the latest Energy Control Catalogue. This company specialises in 6502 based products. Address is P.O. Box 6502, Goodna, 4300. Ph. 07 288 2455. Daneva in Victoria are offering specials on 5.25" and 3" drives at $180 inc. tax. Address is P.O. Box 114, Sandringham 3191. Ph. 03 598 5622. The Queensland Computer Expo is on November 8-11th at the Crest International Hotel. The meeting closed at 5pm.

APPLE DISK II INTERFACE



KAOS-W.A.

One of the topics discussed at our last meeting was the many xx 80 type printers. I had a problem with a "sticky" label which became stuck on the metal plate underneath the platten of my Amust DT80. I contacted the firm I had bought the machine from and was told the only way to fix it was to pull the whole platten assembly to pieces. This would take 1-2 hours at $35 per hour.

next page please

WANTED

A copy of the 'HEX EDIT' text editor. Will supply a disk with Hexdos 4 on it. Please ring and ask for Mark.

FOR SALE

Superboard in case with Tasan video board and Tasker buss. 1 x 8K RAM board (populated), 1 x 8K and 1 x 24K (6116 type) RAM boards (unpopulated), disk controller board. Some documentation and programs on tape including WP65Ø2, games, 65Ø2 disassembler, assembler/editor and extended monitor. Microline 8Ø printer. 5 volt power supply.
Price $75Ø.ØØ ono. Phone

C1P series 1 in OSI case with 16K RAM by piggy-back method. $125.ØØ including postage anywhere in Australia.
E. Richardson,

I am the ex-owner of an OSI C1P and I have left over the Grafix SEB-1 graphics expansion board for the C1P. The specs for the board are: High resolution colour graphics/memory expansion board suitable for connection to C1P/Superboard II. Eleven software selectable display modes from a 32x16 alphanumeric to a 256x192 point addressable display, up to 8 colours, 6K display RAM, RF or composite video output, 16K user expansion RAM, 8 bit parallel port w/handshake and 2 16 bit timers/counters. The board has not been filled but I have all the required chips. If any member of KAOS is willing to make a reasonable offer I would be only to happy to let this board go to a new home.
C.A. Bear,

HELP WANTED

I have a C2 with a serial terminal. As you can imagine, it isn't a lot of fun for graphics/arcade games. There is also no editor for BASIC lines. One continual frustration is with OSI's 65D input routine. SHIFT O causes the cursor to move forward on the terminal. The terminal has direction keys, and $15 is the code needed to backspace the terminal. Would it be possible to modify the input routine to achieve this?

Not happy with this I took the platten assembly out and asked one of the electronic technicians at work to see if he could find an easier way. Within 1/2 an hour he returned it to me, without the offending sticky label. He had pushed out a pin on the right hand end of the platten using a special tool. So to any one else who may have this problem, be assured that there is an easy solution.

Our last alternative meeting was held during August but due to the short notice only 3 people turned up. Unfortunately Gary Giles, whose home the meetings were to be held at has been transferred to Kalgoolie, so future even month meetings are to be held at Peter Van der Weddens home, 24 Parsons Way, Innaloo. Next meeting dates:
Sunday 18th November at Guild House 56 Kishorn Rd, Mt Pleasant at 2pm.
Sunday 16th December at 24 Parsons Way, Innaloo at 2pm.

Gerry Ligtermoet

Registered by Australia Post
Publication No. VBG4212

If undeliverable return to
KAOS, 10 Forbes St
Essendon, Victoria 3040

KAOSKAOS
K Postage K
A Paid A
O Essendon O
S 3040 S
KAOSKAOS